

**Corso di Programmazione
Modulo di Laboratorio
A.A. 2017/2018**

Progetto

Regole per lo svolgimento:

- Il progetto deve essere svolto **singularmente o in gruppi di 2 persone**.
- Il progetto deve essere consegnato **almeno 7 giorni prima** della prova orale di Laboratorio di Programmazione relativa all'appello in cui si desidera registrare l'esame.
- Il progetto deve contenere le classi, i campi e i metodi richiesti **rispettandone esattamente il nome, il tipo e l'ordine dei parametri formali, ed il tipo di ritorno.**
- Nel file allegato è presente la specifica, per ***qualcuna delle classi richieste***, dei campi e dei metodi che bisogna implementare.
Le altre classi devono essere opportunamente aggiunte da voi nella consegna. Trattandosi di classi che estendono altre classi presenti nel file, la loro specifica si deduce in modo inequivocabile.
- Si tenga presente che **non sono sempre presenti** tutti i modificatori (***public, private, protected, static, final, ...***) che devono essere opportunamente aggiunti da voi nella consegna.
- I **metodi accessori e modificatori** non sono tutti presenti nella specifica. Deve essere vostra cura aggiungerli in modo opportuno.
- Si è naturalmente liberi di sviluppare metodi aggiuntivi di supporto ai metodi richiesti, laddove lo si ritenga utile.
- Deve essere consegnato via email all'indirizzo luca.moscardelli@unich.it un file compresso contenente tutti i file sorgenti e avente come nome
[java] progetto 2017_2018 Cognome1 Cognome2.zip
l'email deve avere come oggetto
[java] consegna progetto programmazione 2017-2018 Cognome1 Cognome2
dove Cognome1 e Cognome2 sono i cognomi degli studenti che consegnano; chiaramente Cognome2 va omissso se la consegna è effettuata da uno studente singolo.
In caso di consegna in gruppo la consegna deve essere effettuata da uno dei componenti del gruppo, mettendo in CC nell'email anche l'indirizzo dell'altro componente.

Testo del progetto:

Sviluppare in **Java** un package **insiemistica** per la gestione di insiemi numerici (contenenti numeri interi) e di espressioni che coinvolgano insiemi numerici.

Il package deve contenere le seguenti classi, tutte pubbliche:

1. **Insieme**

Gestisce un insieme di numeri interi.

Nella classe Insieme bisogna aggiungere i campi di istanza che si occupano di rappresentare in memoria un insieme (potete utilizzare un array o un ArrayList per memorizzare gli elementi di un insieme).

2. **Exp (classe astratta)**

Tipo ricorsivo per rappresentare espressioni la cui valutazione è un Insieme. Tale classe astratta ha le seguenti classi che la estendono per rappresentare le espressioni indicate tra parentesi:

- o **ExpInsieme** (il caso base, l'espressione formata semplicemente da un insieme),
- o **ExpUnione** (unione tra le valutazioni A e B di due **Exp e1** ed **e2**),
- o **ExpIntersezione** (intersezione tra le valutazioni A e B di due **Exp e1** ed **e2**),
- o **ExpDifferenza** (differenza tra le valutazioni A e B di due **Exp e1** ed **e2**),
- o **ExpDifferenzaSimmetrica** (la differenza simmetrica tra le valutazioni A e B di due **Exp e1** ed **e2**).

3. **BoolExp (classe astratta)**

Tipo ricorsivo per rappresentare espressioni la cui valutazione è un valore booleano. Tale classe astratta ha le seguenti classi che la estendono per rappresentare le espressioni indicate tra parentesi:

- o **BoolExpTrue** (caso base, l'espressione che vale sempre *true*),
- o **BoolExpFalse** (caso base, l'espressione che vale sempre *false*),
- o **BoolExpSottoinsieme** (l'espressione che confronta le valutazioni A e B di due **Exp e1** ed **e2** e vale *true* se e solo se A è sottoinsieme di B, oppure A e B hanno gli stessi elementi),
- o **BoolExpUguale** (l'espressione che confronta le valutazioni A e B di due **Exp e1** ed **e2** e vale *true* se e solo se A e B hanno gli stessi elementi),
- o **BoolExpAppartenenza** (l'espressione che vale *true* se e solo se un **int k** appartiene alla valutazione A di ad una **Exp e**)
- o **BoolExpAnd** (l'AND di due espressioni booleane **BoolExp bE1** ed **bE2**)
- o **BoolExpOr** (l'OR di due espressioni booleane **BoolExp bE1** ed **bE2**)
- o **BoolExpNot** (il NOT di una espressione booleana **BoolExp bE**)

Note:

- Il file compresso allegato con una parte dei sorgenti java contenenti ulteriori specifiche costituisce parte integrante di questo documento.
- Tutte le classi (tranne quelle astratte) devono sovrascrivere il metodo **toString()** in modo opportuno
- Non è necessario consegnare il metodo **main** (che voi potete chiaramente utilizzare per i vostri test)
- L'unica libreria del Java che potete usare nei file del package **insiemistica** è **java.util.ArrayList** (oltre naturalmente alle classi del package **java.lang**)